Master's Thesis

Multi-task clustering for stock selection to enhance prediction performance over multi data

Eun Ji Bang

Department of Electrical and Computer Engineering

Computer Science and Engineering

Graduate School of UNIST

2020

Multi-task clustering for stock selection to enhance prediction performance over multi data

Eun Ji Bang

Department of Electrical and Computer Engineering

Computer Science and Engineering

Graduate School of UNIST

Multi-task clustering for stock selection to enhance prediction performance over multi data

A thesis/dissertation
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Master of Science

Eun-Ji Bang

06/10/2020 of submission

Approved by

Advisor
Kwang-In Kim

Multi-task clustering for stock selection to enhance prediction performance over multi data

Eun-Ji Bang

This certifies that the thesis/dissertation of Eun-Ji Bang is approved.

06/10/2020

signature

Advisor: Kwang-In Kim

signature

Se Young Chun : Thesis Committee Member #1

signature

Seungryul Baek : Thesis Committee Member #2

Abstract

Stock prices are treated as non-stationary and time-variant data, because of generated by interest of various market participants. Also, research on stock prediction methods has conducted only their own individual or whole historical stock data, ignoring the fact that each price of financial instruments has a mutual organic relationship. Although recent research, propose adversarial learning methods to improve the generalization of the predictive model, but how to employ the correlated impacts of multiple dataset still remains an open problem. To solve this problem, we explore how to improve stock prediction performance by exploring multiple data. We introduce multi-task clustering method to incorporate highly correlated individual stock price data. The proposed method has extensively experimented with actual financial instrument data. Our novel methods outperform the previous state-of-the-art method with an average 1.66% improvement with respect to accuracy.

Contents

List of Figures

List of Tables

CHAPTER I

Introduction

Predictions and analysis of stock price determine the value of cooperation and various financial instruments. The understanding broad stock market is an important parts to both global economy and the growth of the overall industry. The finance market participants which is both investors and industry are interested in the nature value of stock. And they originally want to know whether stocks go up or down after a specific time. It's cruel not only to raise funds to expand their business, but also to let the market profound as liquidity provider. However, it is considered one of the most difficult issues because the features of stock prices and indices are noisy and non-stationary. [1, 2]. There have been many studies on many theoretical and experimental challenges.

Most conventional quantitative trading methods are based on historical transaction data such as price and volume. The most important of these is the Efficient Market Hypothesis (EMH) [3], the hypothesis assumes that in an efficient market, the stock market price fully reflects available information about the market and its components, so there is an opportunity to get excessive profit cases. However, EMH assumes that all investors are aware of all available information in exactly the same way, and no one can achieve greater profitability than other investors with the same amount of investment funds under an efficient market hypothesis. Despite the theoretical wide acceptance of EMH, numerous studies have attempted to disprove the effective market hypothesis experimentally, and empirical evidence has revealed that the stock market is predictable. [4–6].

Traditional approaches to time series prediction use parametric statistical models such as Auto Regressive Moving Average (ARMA), Auto Regressive Integrated Moving Average (ARIMA) and vector automatic regression to find the best estimates. [7–10]. They concluded that the ARIMA model has special potential in short-term forecasting. These quantitative economic models are convenient for explaining and evaluating the relationship between variables by statistical inference, but there are some limitations in financial time series analysis. First, they can't capture the nonlinear nature of the stock price because they assume a model structure of a linear form. Also, it is assumed that the time series is noisy and has time-varying variability, while the variance is constant. [11, 12].

There have been many attempts to solve nonlinear relationships in financial instruments using computer science fields. Recently, neural networks have been shown to be remark performance in predicting future stock price. Many researches and applications of neural networks have demonstrated their merits with respect to classical methods. Also, deep neural networks (DNNs) outperform conventional methods [13–15]. Many successful applications have shown that DNN can be a useful technique for predicting stock prices because the underlying relationships are unknown or difficult to describe, but can capture subtle functional relationships between empirical data.

In recent studies, long short-term memory (LSTM) network, which is appropriately structured to learn temporal patterns, is extensively utilized for various task of time-series analyses [16]. LSTM is advantageous over the conventional Recurrent neural nets (RNNs) as it overcomes the problem of vanishing gradients and as it can effectively learn long-term dependencies through memory cells and gates. If data are given, suitable data-dependent patterns are automatically detected within data proposed an LSTM-based system to predict stock returns and tested it in the U.S and Chinese stock market [17, 18]. They used historical price data of the stock and market indexes for sequential learning. Numerical results confirmed a promising predictive power of LSTMs, which result in an improvement of forecasting accuracy than before [19, 20]. For financial time series analyses using deep neural networks (DNNs), we should be concerned about the problem of over-fitting as data are not sufficient [21, 22]. In a year, the number of data points that we could collect on a daily basis was only approximately 252. DNNs are promising approaches with enhanced representation power as they learn highly complex nonlinear relationships between variables [23, 24].

1.1 Statement of the problem

However, previous methods proposed training historical data of individual stock itself or all stocks to enhance prediction performance [17, 25, 26]. But the stock markets are driven by behavior of various market participants, as perhaps the most important of many variables influencing price [27]. For these reasons, it's not reasonable to utilize limited methods. To overcome this issue, studies have conducted with financial news data [28–30]. Also, there is another way to clarify how stock markets move organically and affect mutually. In other words, stock price movements are results of multiple factors such as macro-economy, financial situation of a company, investors' sentiments, etc. And financial time series contain high noise. To predict stock price movement, features containing useful information are needed, so feature extraction and selection play significant roles in stock price movement prediction. By doing this, we are able to set those datasets into features to train proposed model. One of the novel method is clustering to link each stock as a close neighbor. Thus, it is to investigate the correlation structure within U.S stock exchange (NYSE) and obtain hierarchical structures with dendrograms based on the correlations among individual stocks.

1.2 Aim of Research

The objects of research are to investigate quantifying cross-correlation in terms of supporting the performance of individual stock predictions, as well as understanding the collective behavior between components of a complex system.

1.3 Organization of This thesis

This thesis is described as follows. Chapter II describes our proposed methods, Multi-task clustering for stock selection to enhance prediction performance over multi data. First, we reviews related work and backgrounds. Second, we show our experiments settings such as dataset, baselines, evaluation methods. After that, we explain the details of our proposed methods which is consisting of portfolio under clustering methods and results qualitatively and quantitatively. Finally, in chapter III concludes this thesis with a summary and future work.

CHAPTER II

Multi-task clustering for stock selection to enhance prediction performance over multi data

2.1 Related Work

2.1.1 AutoRegressor Integrated Moving Average (ARIMA)

An autoregressive integrated moving average model (ARIMA) is a form of regression analysis that gauges the strength of one dependent variable relative to other changing variables. The model's goal is to predict future securities or financial market moves by examining the differences between values in the series instead of through actual values.

An ARIMA model can be understood by outlining each of its components as follows:

- **Autoregression** (AR) refers to a model that shows a changing variable that regresses on its own lagged, or prior, values.
- **Integrated** (I) represents the differencing of raw observations to allow for the time series to become stationary, i.e., data values are replaced by the difference between the data values and the previous values.
- **Moving average** (MA) incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

Each component functions as a parameter with a standard notation. For ARIMA models, a standard notation would be ARIMA with p, d, and q, where integer values substitute for the

parameters to indicate the type of ARIMA model used. The parameters can be defined as:

- **p** the number of lag observations in the model; also known as the lag order.
- **d** the number of times that the raw observations are differences; also known as the degree of differencing.
- **q** the size of the moving average window; also known as the order of the moving average.

In an autoregressive integrated moving average model, the data are differenced in order to make it stationary. A model that shows stationarity is one that shows there is constancy to the data over time. Most economic and market data show trends, so the purpose of differencing is to remove any trends or seasonal structures.

2.1.2 Recurrent Neural Networks (RNNs)

Traditional neural networks lack the ability to address future inputs based on the ones in the past. For example, a traditional neural network cannot predict the next word in the sequence based on the previous sequences. However, a recurrent neural network (RNN) most definitely can. Recurrent Neural networks, as the name suggests are recurring. Therefore, they execute in loops allowing the information to persist. In fig. 2.1.2, we have a neural network that takes the

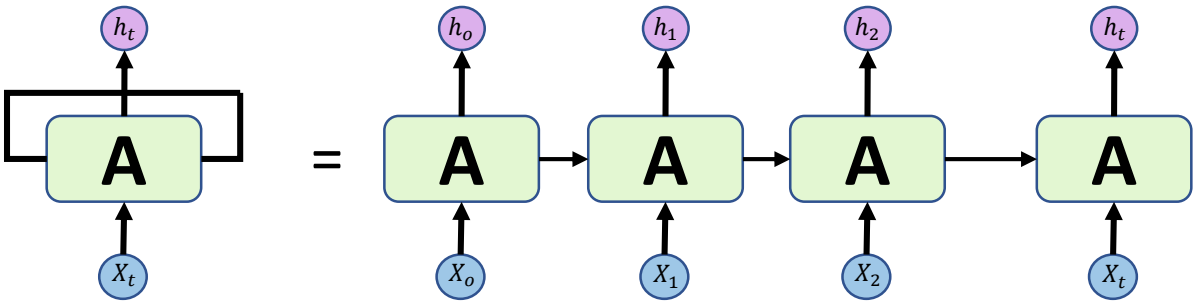


Figure 2.1: Structure of Recurrent neural networks

input x_t and gives use the output h_t. Therefore, the information is passed from one step to the successive step. This recurrent neural network, when unfolded can be considered to be copies of the same network that passes information to the next state. RNNs allow us to perform modeling over a sequence or a chain of vectors. These sequences can be either input, output or even both.

2.1.3 Gated Recurrent Units (GRUs)

Gated recurrent unit (GRUs) is part of a specific model of recurrent neural network that intends to use connections through a sequence of nodes to perform machine learning tasks

associated with memory and clustering, for instance, in speech recognition. Gated recurrent units help to adjust neural network input weights to solve the vanishing gradient problem that is a common issue with recurrent neural networks. The key idea of GRUs is that the gradient

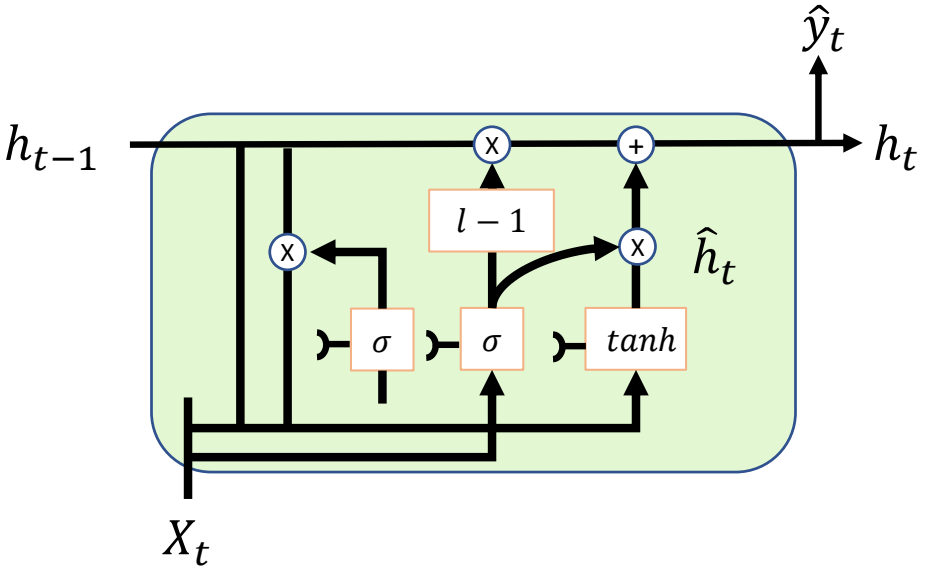


Figure 2.2: Structure of Gated Recurrent Units, fully gated version

chains do not vanish due to the length of sequences. This is done by allowing the model to pass values completely through the cells. The architecture of GRU is described in fig. 2.1.3. The model is defined as the following:

$$
\begin{aligned}
z_t &= \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1} + b^{(z)}\right) \\
r_t &= \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1} + b^{(r)}\right) \\
h_t &= \tanh\left(W^{(h)}x_t + U^{(h)}h_{t-1} \circ r_t + b^{(h)}\right) \\
h_t &= (1^{\infty}z_t) \circ h_{t^*1} + z_t \circ \tilde{h}_t
\end{aligned}
$$

$\circ$ is used as the Hadamard product, which is just a fancier name for element-wise multiplication. $\sigma(x)$ is the Sigmoid function which is defined as $\sigma(x) = \frac{1}{1+e^{-x}}$. Both the Sigmoid function and the Hyperbolic Tangent function (tanh) are used to squish the values between 0 and 1.

z_t functions as a filter for the previous state. If z_t is low (near 0), then a lot of the previous state is reused. The input at the current state x_t does not influence the output a lot. If z_t is high, then the output at the current step is influenced a lot by the current input x_t, but it is not influenced a lot by the previous state h_{t-1}. r_t functions as forget gate (or reset gate). It allows the cell to forget certain parts of the state.

2.1.4 Long Short Term Memory networks (LSTMs)

Long Short-Term Memory (LSTM) networks are a modified version of Recurrent Neural Networks (RNNs), which makes it easier to remember past data in memory. Recurrent Neural Network is a generalization of feed-forward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network. However, RNNs suffer from Gradient vanishing and exploding problems. Also, it cannot process very long sequences if using tanh or relu as an activation function. The vanishing gradient problem of RNN is resolved. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using back-propagation. On LSTM networks, three gates are present (see fig.2.1.4):

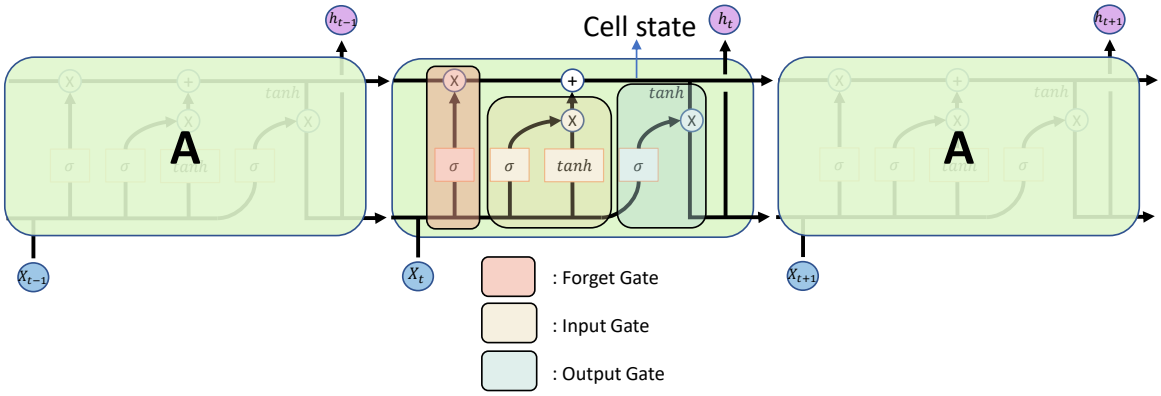


Figure 2.3: LSTM gates

- **forget gate** shows what details to be discarded from the block. It is decided by the sigmoid (σ) function. It looks at the previous state (h_{t-1}) and the content input (X_t) and outputs a number between 0 (discard) and 1 (keep) for each number in the cell state C_{t-1}. $f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$

- **Input gate** discovers which value from input should be used to modify the memory. Sigmoid function decides which values to let through 0,1. And tanh function gives weight to the values which are passed deciding their level of importance ranging from -1 to 1.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right)$$

- **Output gate** is that the input and the memory of the block is used to decide the output. And tanh function gives weight to the values which are passed deciding their level of importance ranging from -1 to 1 and multiplied with output of sigmoid.

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh(C_t)$$

LSTMs is a network where cell state and hidden state are recursively obtained. Therefore, the gradient of the cell state and the gradient of the hidden state are affected by the gradient value of the previous point. LSTMs is well suited to capture sequential information from temporal data and has shown advantages in machine translation, speech recognition, and image captioning etc.

2.1.5 Correlation and Covariance between individual stocks

Correlation is a statistical technique for measuring and describing the relationship between two variables. There are examples that the two variable weights, X_1, and length, X_2. Screening this data indicates that there is a relationship between the measured quantities, assuming the population is reasonable. A taller person is usually heavier than a shorter person. Two statistical measurements that help to assess the relationship between two random variables are covariance and correlation [31]. The covariance can be defined as:

$$COV(X_1, X_2) = E[(X_1 - \mu_1)(X_2 - \mu_2)] \tag{II.1}$$

where $\mu_1 = E(X_1)$, i.e. the expected value of X_1, and $\mu_2 = E(X_2)$. If X_2 tends to be large when X_1 is large and small when X_1 is small, then X_1 and X_2 will have a positive covariance. On the other hand, if X_2 is small, X_2 is large, and if X_1 is small, X_2 is large, then X_1 and X_2 Has negative covariance. Covariance measures the associative direction, but the values are unit-specific, making comparison difficult. To measure the association strength independently, the covariance must be normalized with respect to the variance of the measurement variables. The correlation between the two variables reflects the degree to which the variables are related. The most common correlation measure is Pearson's Correlation. The correlation coefficient between the two variables X_1 and X_2 is in this case given by

$$\rho_{i,j} = \frac{COV(X_1, X_2)}{\sqrt{V(X_1)\,V(X_2)}} \tag{II.2}$$

where $V(X_i)$ is the variance of varible X_i. Pearson's correlation reflects the degree of linearity between the two variables. The range is +1 to -1. A correlation of +1 means that there is a complete positive linear relationship between the variables. A correlation of -1 indicates that there is a complete negative linear relationship between the variables, and 0 indicates no correlation.

Joint movement between individual stocks and stocks and certain market indices plays an important role in finance. 2.11 shows a comparison of different stock trends and correlation metrics.

2.1.6 Granger causality

Granger causality (G-causality) is a popular method for studying casual links between random variables [32]. Specifically, suppose that the spike train of neuron i at time bin m can be predicted given the neuron's own firing history and that of another neuron j using the bivariate auto-regressive model:

$$S_i(m\Delta) = \sum_{k=1}^{K} A_{ii}(k) S_i((m-k)\Delta) + \sum_{k=1}^{K} A_{ij}(k) S_j((m-k)\Delta) + \varepsilon_{i|j}(m\Delta) \qquad \text{(II.3)}$$

where K is the maximum number of lags (model order) and A represents the linear regression coefficients obtained by minimizing the squared prediction error $\varepsilon_{i|j}$ when S_j is used to predict S_i. Neuron j is said to Granger-causality neuron if the inclusion of S_i in II.3 reduces the variance of the prediction error.

The null hypothesis is that the y does not Granger causality x. A user specifies the two series, x and y, along with the significance level and the maximum number of lags to be considered. The function chooses the optimal lag length for x and y based on the Bayesian Information Criterion. The function produces the F-statistic for the Granger Causality Test along with the corresponding critical value. We reject the null hypothesis that y does not Granger causality x if the F-statistic is greater than the critical value.

2.1.7 Hierarchical Clustering

HC (Hierarchical Clustering) is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is different from other clusters, and the objects within each cluster are similar. The aim of HC is finding the best step at each cluster fusion (greedy algorithm) which is done exactly but resulting in a potentially sub-optimal solution. In a cohesive or bottom-up clustering method, each observation is assigned to its own cluster. Then HC calculates the similarity (e.g. distance) between each cluster and join the two most similar clusters. Finally, HC repeats steps 2 and 3 until only one cluster remains. Before any clustering is performed, it is required to determine the proximity matrix containing the distance between each point using a distance function. Then, the matrix is updated to display the distance between each cluster. Details are illustrated in the diagrams below:

In 2.1.7, HC starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps: (1) identify the two clusters that are closest together, and (2) merge the two most similar clusters. This iterative process continues until all the clusters are

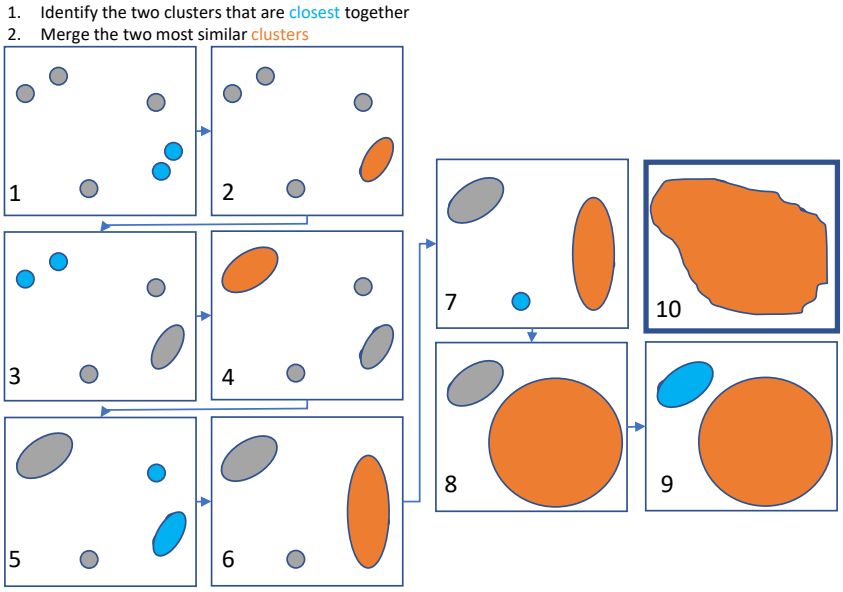


Figure 2.4: Illustration of hierarchical clustering steps

merged together. Finally, The main The main output of Hierarchical Clustering is a dendrogram, which shows the hierarchical relationship between the clusters in 2.5:

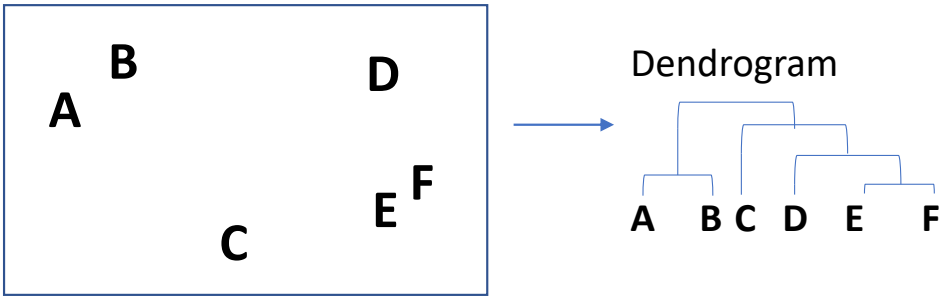


Figure 2.5: Illustration of hierarchical clustering in dendrogram

2.1.8 K-Means Clustering

K-means clustering is a simple unsupervised learning algorithm that is used to solve clustering problems. It follows a simple procedure of classifying a given data set into a number of clusters, defined by the letter "k," which is fixed beforehand. The clusters are then positioned as points and all observations or data points are associated with the nearest cluster, computed, adjusted and then the process starts over using the new adjustments until a desired result is reached.

It is used mainly in statistics and can be applied to almost any branch of study. For example, in marketing, it can be used to group different demographics of people into simple groups that make it easier for marketers to target. Astronomers use it to sift through huge amounts of astronomical data; since they cannot analyze each object one by one, they need a way to

statistically find points of interest for observation and investigation. Given a set of observations

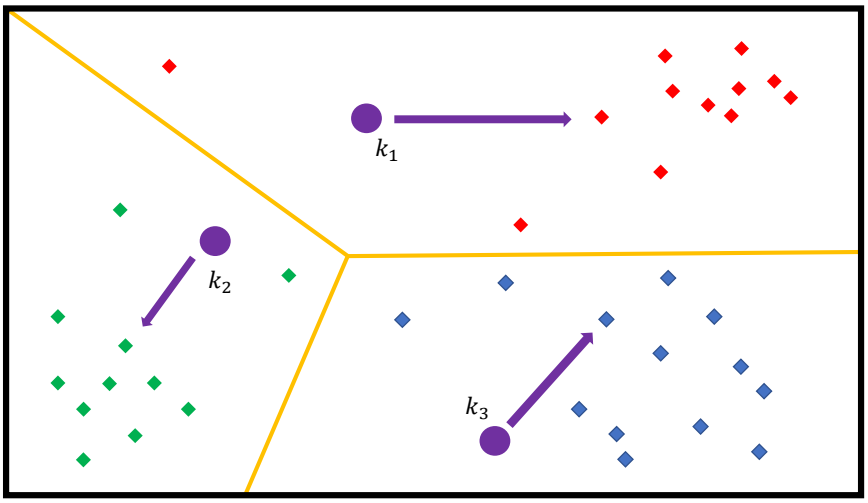


Figure 2.6: Illustration of K means clustering. Each example is assigned to the centroid k_j closest to it. Then, Kmeans calculate the average of the objects assigned to the centroid.

$(x_1, x_2, \ldots, x_n)$, here each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into $k(\leq n)$ sets $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance). Formally, the objective is to find:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i \tag{II.4}$$

where $\boldsymbol{\mu}_i$ is the mean of points in S_i This is equivalent to minimizing the pairwise squared deviations of points in the same cluster:

$$\arg\min_{\mathbf{s}} \sum_{i=1}^{k} \frac{1}{2\,|S_i|} \sum_{\mathbf{x},\mathbf{y} \in S_i} \|\mathbf{x} - \mathbf{y}\|^2 \tag{II.5}$$

The equivalence can be deduced from identity $\sum_{r=S_1} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \sum_{r+r \in S_1} (\mathbf{x} - \boldsymbol{\mu}_i)\,(\boldsymbol{\mu}_i - \mathbf{y})$. Because the total variance is constant, this is equivalent to maximizing the sum of squared deviations between points in different clusters (between-cluster sum of squares), which follows from the law of total variance.

2.1.9 Hilbert-Schmidt Independence Criterion (HSIC)

The Hilbert-Schmidt independence criterion (HSIC), introduced by Gretton et al [33]. HSIC is a useful method for testing if two random variables are independent. The root of the idea is that while $\operatorname{Cov}(A, B) = 0$ does not imply that two random variables A and B are independent,

having $\text{Cov}(s(A), t(B)) = 0$ for all bounded continuous functions s and does actually imply independence [34]. Since going over all bounded continuous functions is not tractable, Gretton et al [33] propose evaluating $\sup_{s\in\mathcal{F},t\in\mathcal{G}} \mathbb{C}$ ov $[s(x), t(y)]$ where $\mathcal{F}, \mathcal{G}$ are universal Reproducing Kernel Hilbert Spaces (RKHS). This allows for a tractable computation and is equivalent in terms of the independence property. Gretton et al. [33] then introduced HSIC as an upper bound to HSIC is a non-parametric method that does not assume a specific noise distribution for ε [33].

Consider two random variables X and Y, residing in two metric spaces $\mathcal{X}$ and $\mathcal{Y}$ with a joint distribution on them, and two separable RKHSs $\mathcal{F}$ and $\mathcal{G}$ on $\mathcal{X}$ and $\mathcal{Y}$ respectively. HSIC is defined as the Hilbert Schmidt norm of the cross covariance operator:

$$\text{HSIC}(X, Y; \mathcal{F}, \mathcal{G}) \equiv \|C_{xy}\|^2_{\text{HS}}$$

Gretton et al. [33] show that:

$$\text{HSIC}(X, Y; \mathcal{F}, \mathcal{G}) \geq \sup_{s\in\mathcal{F},t\in\mathcal{G}} \text{Cov}[s(x), t(y)]$$

We now state Theorem 4 of Gretton et al. [33]) which shows the properties of HSIC as an independence test:

Theorem 1 (Gretton et al. [33], Theorem 4). Denote by $\mathcal{F}$ and $\mathcal{G}$ RKHSs both with universal kernels. k, l respectively on compact domains $\mathcal{X}$ and $\mathcal{Y}$. Assume without loss of generality that $\|s\|_\infty \leq 1$ for all $s \in \mathcal{F}$ and likewise $\|t\|_\infty \leq 1$ for all $t \in \mathcal{G}$ Then the following holds: $\|C_{xy}\|^2_{HS} = 0 \Leftrightarrow XY$ Let $\{(x_i, y_i)\}_{i=1}^n$ be i.i.d. samples from the joint distribution on $\mathcal{X} \times \mathcal{Y}$. The empirical estimate of HSIC is given by:

$$\widehat{\text{HSIC}}\left(\{(x_i, y_i)\}_{i=1}^n; \mathcal{F}, \mathcal{G}\right) = \frac{1}{(n-1)^2} \operatorname{tr} KHLH$$

where $K_{i,j} = k(x_i, x_j), L_{i,j} = l(y_i, y_j)$ are kernel matrices for the kernels k and l respectively, and $H_{i,j} = \delta_{i,j} - \frac{1}{n}$ is a centering matrix. The main result of Gretton et al. [8] is that the empirical estimate $\widehat{\text{HSIC}}$ converges to HSIC at a rate of $O\left(\frac{1}{n^{1/2}}\right)$, and its bias is of order $O\left(\frac{1}{n}\right)$

2.2 Background

2.2.1 Adv-ALSTM

We follow Adversarial Attentive LSTM (Adv-ALSTM) [35] as background. The main contributions of the proposed method are that investigation of generalization difficulty of stock prediction. They suggest that adversarial learning reaches more generalizable and robust. They adapt that Adversarial Perturbation (AP) is the direction that leads to LSTMs with an attention method to increase performance of stock forecasting and the largest changes in the model prediction. Adversarial training is proposed to account for the stochastic property of the stock market to learn stock movement prediction model. Their adversarial method for the Attentive LSTM model is an expressive model for temporal data. When the proposed method adds perturbations to the prediction features in last hidden layer, it is possible to optimize the perturbations to make them draw a decision boundary from the model's output as much as possible.

Before we explore Adv-ALSTM methods, they define the predictive function to set the formula for stock movement forecasting operations as $\hat{y}^s = f\left(\boldsymbol{X}^s; \boldsymbol{\Theta}\right)$ which maps a stock (s) from its temporal features (X^s) to the label space. In other words, the function f with parameters Θ aims to predict the movement of stock s at the next time-step from the sequential features X^s in the latest T time-steps. $\boldsymbol{X}^s = [\boldsymbol{x}_1^s, \cdots, \boldsymbol{x}_T^s] \in \mathbb{R}^{D \times T}$ is a matrix which represents the sequential input features (e.g., open and close prices, as detailed in Table 2.1) in the lag of past T time-steps, where D is the dimension of features. Assuming that we have S stocks, Adv-ALSTM

Generated features	Formula
open_close, high_close, low_close	e.g., open_close= $\text{open}_t/\text{close}_t$ -1
t_close, t_adj_close	e.g., t_close=$\text{close}_t/\text{close}_{t-1}$ -1
5day, 10day, 15day, 20day, 25day, 30day	e.g., $5-day = \frac{\sum_{i=0}^{4} adj-\text{close}_{t-i}/5}{adj-close_t}$-1

Table 2.1: Generated features the end of day stock price

learn the prediction function by fitting the ground truth labels $\mathbf{y} = \left[y^1, \cdots, y^S\right] \in \mathbb{R}^S$, where $y^s \in (1/-1)$ is the label of stock s in the next time-step. They then formally define the problem as:

Input: given training dataset $\{(\boldsymbol{X}^s, y^s)\}$

Output: A prediction function $f\left(\boldsymbol{X}^s; \boldsymbol{\Theta}\right)$, predicting the movement of stock s in the following time-step.

Instead of directly making prediction from last hidden layer in LSTM, they adapted adversarial examples (AEs). AE is a malicious input created by adding intentional perturbations to the function of clean data. The perturbation named as an AP (Adversarial Perturbation) cannot be applied directly to stock predictions in the direction that brings the largest change in the

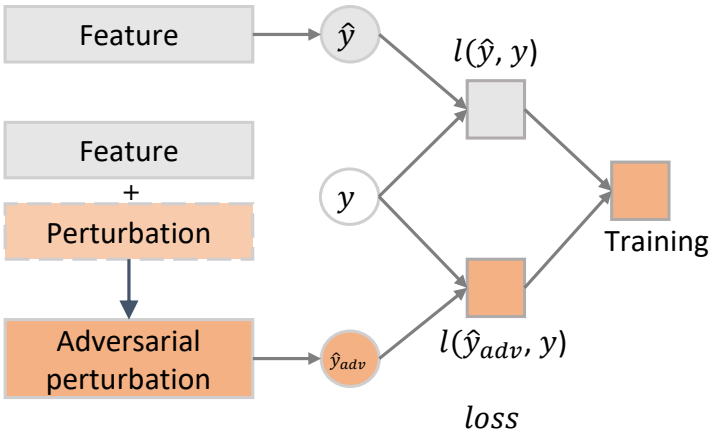


Figure 2.7: A graphical structure of the Adv-ALSTM (baseline).

model prediction. (See Fig. 2.7). Perturbation calculations can be time consuming as they rely on the calculation of gradients for inputs (due to back-propagation through the time step of the LSTM layer). Also, given the fact that the gradients of the input depend on different time steps, there may be unintended interactions between the perturbations of different time steps that cannot be controlled. To solve these problems, they propose to link APs from last hidden layers.

2.3 Proposed method

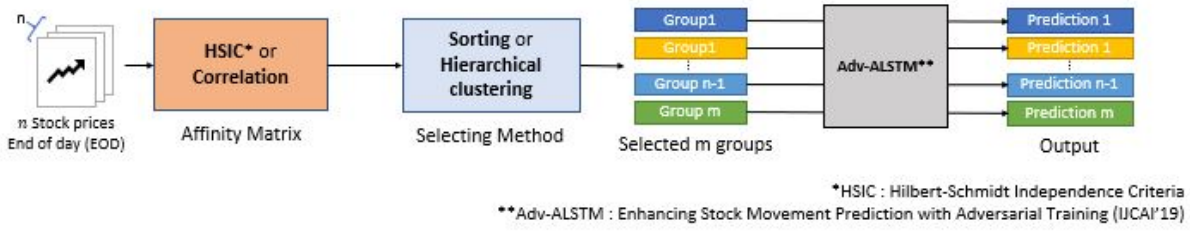


Figure 2.8: Overview of proposed method (Selecting methods).

Fig. 2.8 show that the overall structure of our proposed method. First, we normalize raw data, End of day individual stock price with z-score [36]. Normalization is standardizing each data or changing it to make it easier to compare with others. Second, we adopt 2 types affinity matrices with HSIC and Pearson correlation methods. In addition, we propose Selecting Method to group related stocks using choosing top s neighbors or Hierarchical clustering.

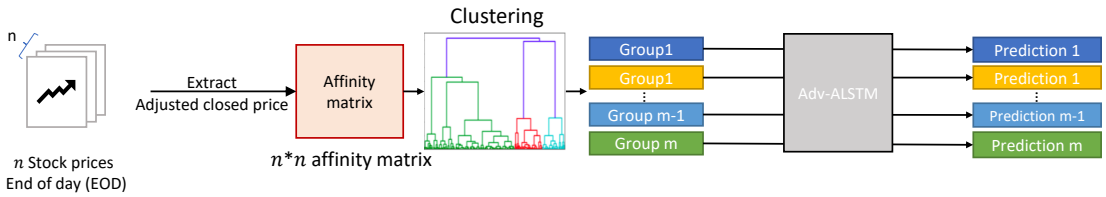
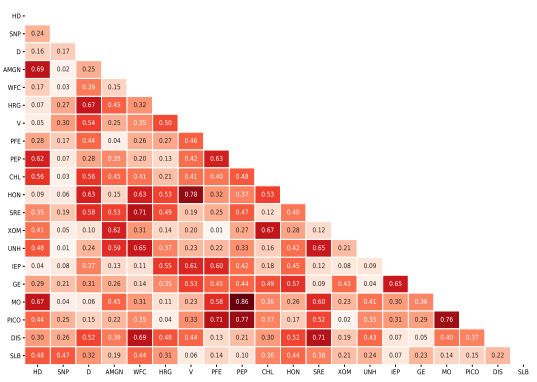


Figure 2.9: Architecture of Hierarchical clustering (HC) affinity matrix (Correlation, HSIC)

2.3.1 Clustering - Hierarchical clustering (HC) affinity matrix (Correlation, HSIC)

We generate distance matrix based on both Pearson correlation coefficient and HSIC is used [37]. And we demonstrate correlations method for stock adjusted closed price in Fig. 2.10.

Fig. 2.9 show that structure of affinity matrices methods with Hierarchical clustering (HC). m grouped stocks based on HC are trained as Adv-ALSTM.



	HD	SNP	D	AMGN	WFC	HRG	V	PFE	PEP	CHL
HD	1.00	0.59	0.63	0.88	0.75	0.70	0.64	0.25	0.86	0.66
SNP	0.59	1.00	0.81	0.68	0.81	0.83	0.63	0.43	0.76	0.37
D	0.63	0.81	1.00	0.77	0.89	0.93	0.87	0.69	0.69	0.14
AMGN	0.88	0.68	0.77	1.00	0.77	0.83	0.73	0.45	0.78	0.59
WFC	0.75	0.81	0.89	0.77	1.00	0.87	0.83	0.45	0.83	0.27
HRG	0.70	0.83	0.93	0.83	0.87	1.00	0.86	0.62	0.72	0.29
V	0.64	0.63	0.87	0.73	0.83	0.86	1.00	0.68	0.57	0.10
PFE	0.25	0.43	0.69	0.45	0.45	0.62	0.68	1.00	0.19	0.23
PEP	0.86	0.76	0.69	0.78	0.83	0.72	0.57	0.19	1.00	0.61
CHL	0.66	0.37	0.14	0.59	0.27	0.29	0.10	0.23	0.61	1.00

Figure 2.10: (Left) Correlation matrix over 20 stocks (out of 87), (Right) An example of correlation matrix

Also we compare and plot graphs to check the correlation matrix classifies relationships of each stock trend. In addition, we identify which stocks belong to which industries. The correlation coefficient between the stocks in fig. 2.11 is determined over the 420 trading days of 2 years using (see in fig. 2.10 left). The result agrees with the visual impression of the compared time series. The difference in co-movement is hardly surprising, given the fact that both Depot (HD) and Amgen (AMGN) main business activity is consumer goods and healthcare, respectively. Also many studies show that both consumer goods and healthcare sectors (especially, pharma) have been high correlated [38]. Therefore, it is likely to be affected by cross-industry factors. However, HD (Home Depot) and PCLN (Booking Holding) belong to the same industry, so they are less likely to make similar moves. The correlation coefficient reflects only the degree of linear

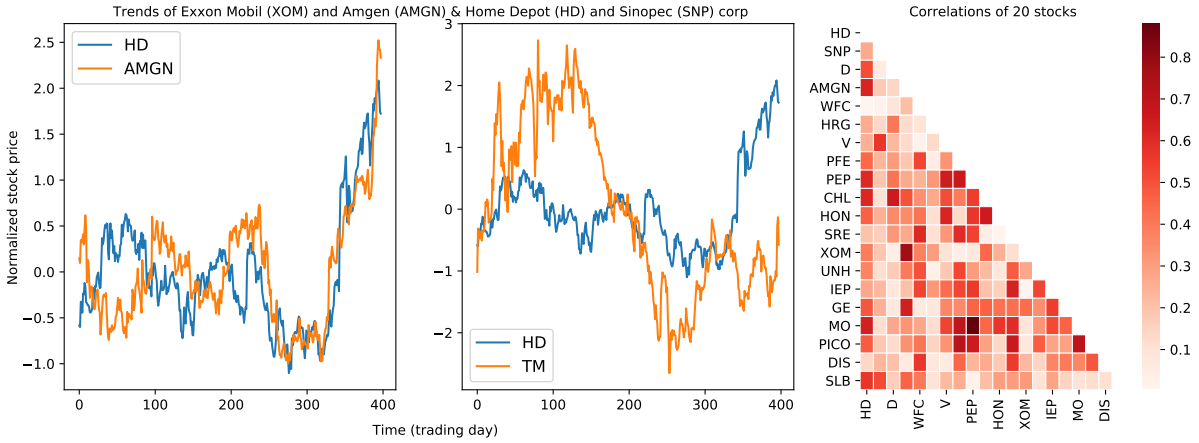


Figure 2.11: A comparison between the daily (normalized with z-score, reflected 87 samples of mean) adjusted closed price for Home Depot (HD)/Amgen (AMGN) (left) and Home Depot (HD) / Booking Holding (PCLN) (right) during 2013.01 2̃014.12. The correlation of HD AMGN and MO PFE are 0.63 and -0.26, respectively.

relationship between the movements of two different stocks. The information of this matrix can be explained by many possible reasons. First, industries producing the same inventories tend to form groups, because they are competitors and affected by the same environment in the stock exchange. And the existence of complementary among firms, IT industries can be a clear example. Intel develops processors used by computers designed by IBM and HP, while they use programs designed by Microsoft.

Industry	Acc	F1	Mcc
Information Technology	0.5444	0.4949	0.1216
Healthcare	0.5501	0.4658	0.1188
Utilities	0.5659	0.6348	0.1288
Concumer	0.5313	0.4595	0.1186
Industrials	0.5500	0.4362	0.1130
Financials	0.5844	0.5238	0.1583
Energy	0.5643	0.2878	0.1271
Telecommunication Services	0.6033	0.4000	0.2180
Average performace	0.5617	0.4629	0.1380

Table 2.2: Performance of the same industries. the average performance underperform Adv-ALSTM method (0.572, ACC)

However, stocks in the same industry group are not always correlated (see. Fig. 2.11). This is because the current classification of industries is not perfect and is still in the research phase [39]. Moreover, we select stocks between the same industries to check performance in table 2.2.

Because it is needed to consider identifying appropriate groups of stocks, we adapt Hierarchical clustering (HC) methods. HC do initial clustering and construct a dendrogram, where the

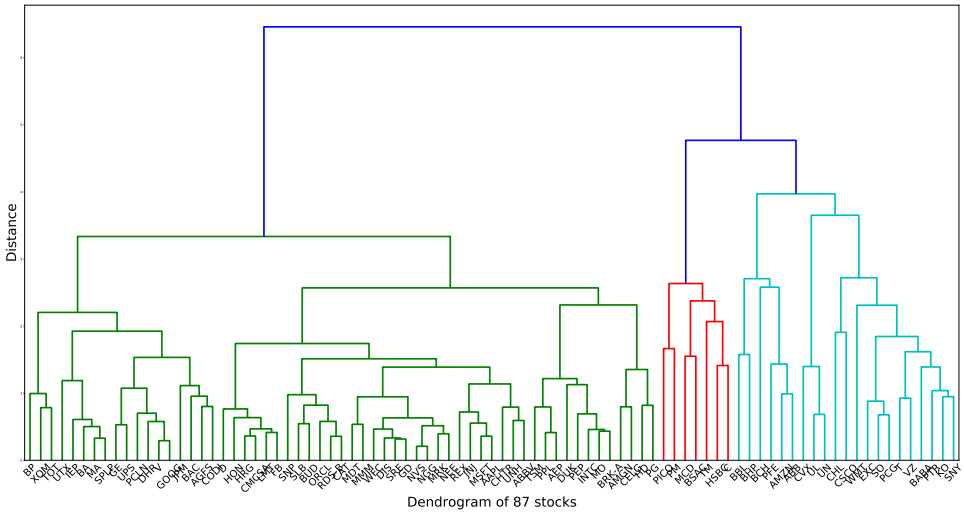


Figure 2.12: Dendrogram of Hierarchical clustering results

centroid clustering is used and the similarity is computed by the Euclidean distance between features. Table 2.3 shows the results of HC at 3 cluster. We also compare the performance of HC quantitatively and qualitatively.

Fig. 2.12 is illustrated each of stock distance on dendrogram. We can easily understand how each stock has relationships. Fig. 2.13 show that how well HC clustered. We select and plot several stocks to visualize the performance of clusters. It's obvious stocks in the same industry are highly correlated and has similar trends. But stocks in different industries are less correlated than same industry. We show that the HC is well-performed. The need to calculate the distance

	Stock list
Cluster 1 (skyblue)	**BP, XOM,TOT**, UTX,IEP, BA, MA,SPLP,GE, UPS, PCLN, DHR, V, GOOG, JPM,BAC,AGFS,CODI,D,HON,HRG,CMCSA,LMT,FB,SNP,SLB,BUD,ORCL, RDS-B,CAT,MDT,MMM,WFC,DIS,SRE,GD,NVS,NGG,MRK,NEE,REX,JNJ ,**MSFT,AAPL,CHTR,UNH**,ABBV,TSM,PPL,AEP,DUK, PEP,INTC,MO, BRK-A,AMGN,CELG,HD,PG
Cluster 2 (pink)	PICO,PM,MCD,BSAC,TM,HSBC
Cluster 3 (yellow)	C,BBL,BHP,BCH,PFE,AMZN,ABB,CVX,UL,UN,CHL,CSCO,**WMT**, EXC,SO,PCG,T,VZ,BABA,PTR,KO,SNY

Table 2.3: Result on hierarchical clustering. See fig. 2.12 to match colors in this table index

between different stocks in the financial market has been mentioned. We adopt the dendrogram constructed into m groups. This multivariate analysis method is designed to extract information about the number of key factors that characterize the dynamics of the investigated system and the composition of groups in which the market is essentially organized.

We apply the groups into multi-task learning to a problem of stock prediction. We consider

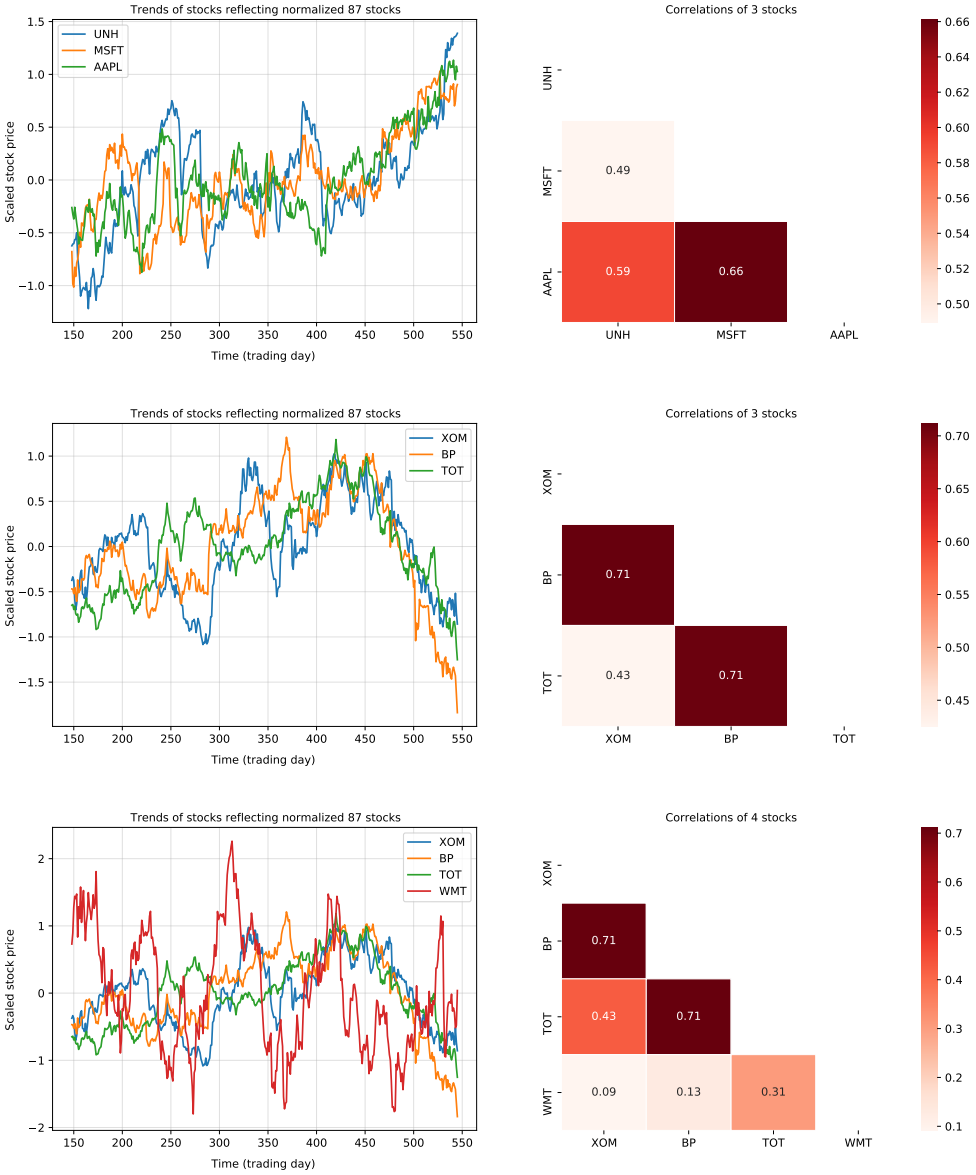


Figure 2.13: (Top) trends of UNH, MSFT, APPL in cluster 1, (Middle) trends of XOM, BP, TOT in cluster 1, (Bottom) Comparison of farthest distance from HC

87 assets, listed in S&P500 from the NYSE. Finally, we plug into our baseline algorithms, Adv-LSTM.

2.3.2 Selection top s neighbors from affinity matrix (Correlation, HSIC)

We also introduce another selecting method which is selection top s neighbors from an affinity matrix using correlation or HSIC and utilize adjusted closing price from end of day with n stocks.

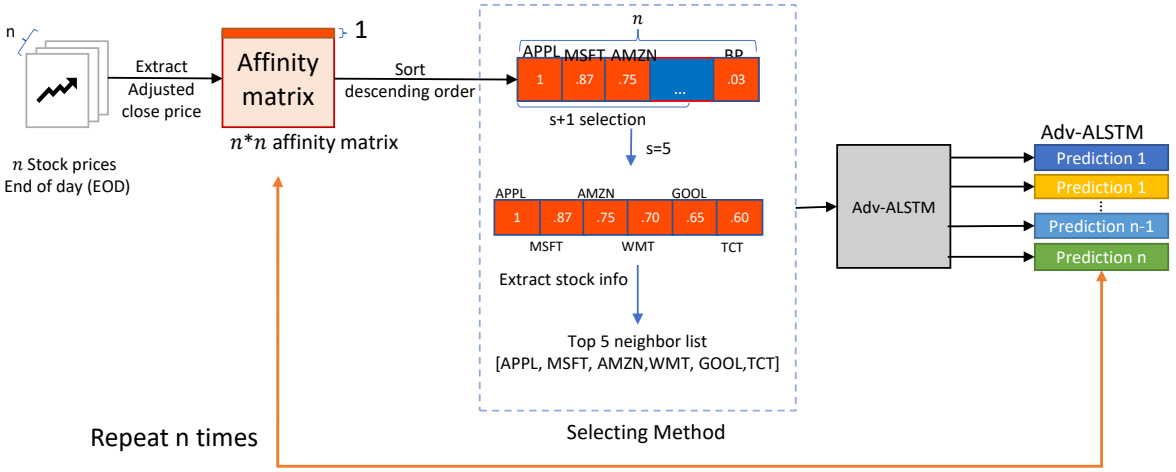


Figure 2.14: Architecture of choosing n neighbor affinity matrix

Adjusted closing price amends a stock's closing price to accurately reflect that stock's value after accounting for any corporate actions such as dividends and distributions and rights offerings etc. It is considered to be the true price of that stock and is often used when examining historical returns or performing a detailed analysis of historical returns. We generate affinity matrices with correlation or HSIC and select a stock for sorting descending order. After that, our proposed method chooses $s+1$ stocks. In addition, we experiment with additional method based on affinity neighbor. Fig. 2.14 show that process of affinity neighbor method. This is a technique to search for a stock. After sorting the correlation with the stock in descending order, select n with high correlation. Selected n stocks are used for learning together and the method is repeated 87 times. For example, we extracted 5 stocks (Microsoft, Amazon, Britsh petroleum, Walmart, ACE) that have a high correlation with Apple, the first stock, and trained them together. This method also outperforms Adv-ALSTM model at HSIC matrix selected top 40 high correlated neighbors. The details are described in table 2.7.

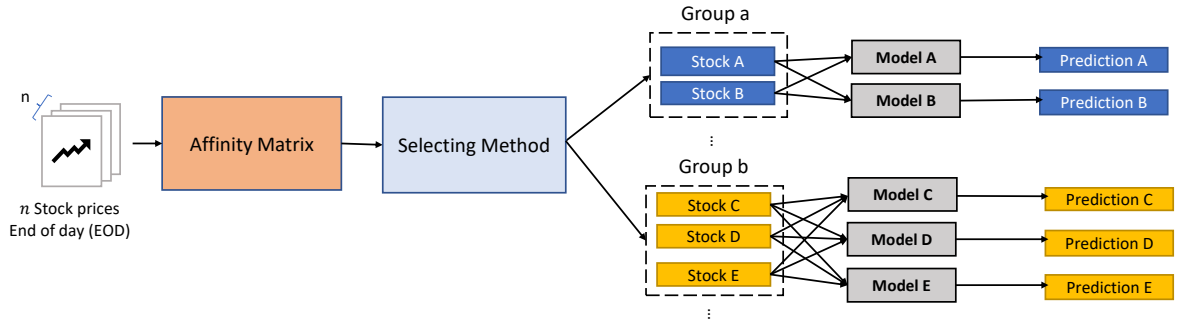


Figure 2.15: Training methods

Fig. 2.15 is described as a sequence of our training methods. Note that each stock has their weight and share the weight parameters in a same group.

2.3.3 Loss function

The baseline, Adv-ALSTM has objective function $\mathbf{\Gamma}$ as follow:

$$\sum_{i=1}^{S} l\left(y^s, \hat{y}^s\right) + \frac{\alpha}{2}\|\Theta\|_F^2, l\left(y^s, \hat{y}^s\right) = \max\left(0, 1 - y^s \hat{y}^s\right)$$

The first term is a hinge loss, which is widely used for optimizing classification models (more reasons of choosing it is further explained at the end of the section). The second term is a regularized on the trainable parameters to prevent overfitting. However, we know that simple hinge loss leads convergence and performance than proposed by baseline loss function with various experiments. Thus, we adopt hinge loss in all experiments.

2.4 Experiments

2.4.1 Experimental Settings

Dataset We also follow Feng [35] dataset to compare performance results exactly. We adopt to predict the Standard & Poor's 500 (S&P 500) index and its individual stocks listed on the New York Stock Exchange (NYSE) from ACL18 [40]. ACL18 contains 87 high-trade-volume in U.S stock market. Detailed statistics of training, development (tuning) and test sets are shown in table 2.4. Also, ACL18 is end of day (EOD) data and each stock has 5 open, high, low, close,

	Train	Validation	Test
# of day	654	77	121
#of positive/negative GT	331 / 323	46 / 31	64 / 57
Time interval	Jan.01.2014 ∼Aug.01.2015	Aug.02.2015 ∼Oct.02.2015	Oct.02.2015 ∼Jan.01.2016

Table 2.4: Explanation of S&P500 dataset. There are 87 assets removed weekends, public holidays and lack historical prices. Also, we adapt Feng's split train, validation and test ways.

volume, adjusted close price (OHLCV). We only use adjusted closing price to generate affinity matrices. But, we utilized all columns to build the model (Details are described in table. 2.1).

Baselines We compare prediction performance on same dataset as follow:

- **LSTM** is Long Short Term Memory networks [41]. We tune three hyper-parameters, number of hidden units (U), lag size (T);
- **ALSTM** is the Attentive LSTM [42], which is optimized with normal training. Similar as LSTM, we also tune U, T;

- **Adv-LSTM** is proposed by Feng [35]. It is also same as ALSTM;

Evaluation methods We evaluate the prediction performance with three metrics, Accuracy (Acc), F1 score [43], and Matthews Correlation Coefficient (MCC) [44] of which the ranges are in [0, 100] and [-1, 1]. These methods are used as a statistical measure of how well a binary classification test correctly identifies. The formulas for evaluation are as follows:

- Accuracy is the number of correctly predicted data points out of all the data points. it is defined as the number of true positives and true negatives divided by the number of true positives, true negatives, false positives, and false negatives. A true positive or true negative is a data point that the algorithm correctly classified as true or false, respectively. Accuracy can be formulated as $= \frac{TP+TN}{TP+TN+FP+FN}$
- F1-score is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results returned by the classifier, and r is the number of correct positive results divided by the number of all relevant samples. F1-score can be formulated as $2 \times \frac{1}{\frac{1}{\text{Precision}}+\frac{1}{\text{Recall}}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+ \text{Recall}}$
- MCC takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes.[3] The MCC is in essence a correlation coefficient between the observed and predicted binary classifications; it returns a value between 1 and +1. A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and 1 indicates total disagreement between prediction and observation. MCC can be formulated as $\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

where, TP = True positive; FP = False positive; TN = True negative; FN = False negative; Precision$=\frac{TP}{TP+FP}$; Recall $=\frac{TP}{TP+FN}$.

The higher the value of the metric, the better the performance. Also, we valid our performance by running market simulation.

2.5 Results

2.5.1 Performance comparison

Table 2.5 show the prediction performance of compared methods on test dataset regarding Acc, F1 score and MCC, respectively. When we see table 2.5, there are several observations:

Table 2.5 show the prediction performance of comparing methods on the test dataset regarding Acc, F1 score and MCC, respectively. When we see table 2.5, there are several observations:

HC Adv-ALSTM on 3 number of clusters yield the best performance in Acc and MCC ways. HC Adv-ALSTM improve performance 1.66% and 40.3% on test data with Acc and MCC, respectively. Our proposed method obviously enhances prediction performance in terms of effectiveness of underlying data [45]. In table 2.5, we show that the average test performance when

Methods	Acc	F1 score	MCC	Remarks
LSTM [17]	0.53	0.5124	0.0674±5e-3	
ALSTM [42]	0.54	0.5324	0.1043±7e-3	
Adv-ALSTM [35]	0.5720±—-	0.5542	0.1483±—-	
HSIC adv-ALSTM	0.5767±0.011	**0.5917**	0.1219	Top 40 neighbors from HSIC
HC adv-ALSTM	**0.5825±0.005**	0.5663	**0.1886**	3 number of cluster

Table 2.5: Performance comparison on five different methods

the method performed best in the verification set at 10 different runs. HSIC and HC stand for Hilbert-Schmidt Independence Criteria and Hierarchical Clustering, respectively. Also, our experiments on various numbers of clusters. HC, ends when all clusters are connected. We can control the number of clusters by setting thresholds. Fig. 2.16 are examples dendrograms based on different threshold.

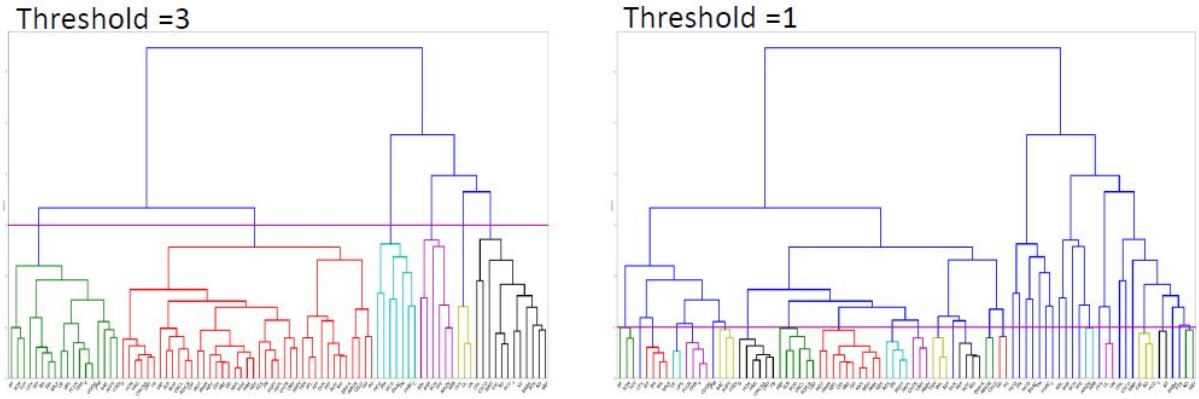


Figure 2.16: Dendrogram of clustering results based on threshold

Most of the performance is yielded on correlation matrix (See in table 2.6). Results of Acc and MCC evaluation on 3 clusters are the best. But cluster 8 outperform other clusters in F1 score.

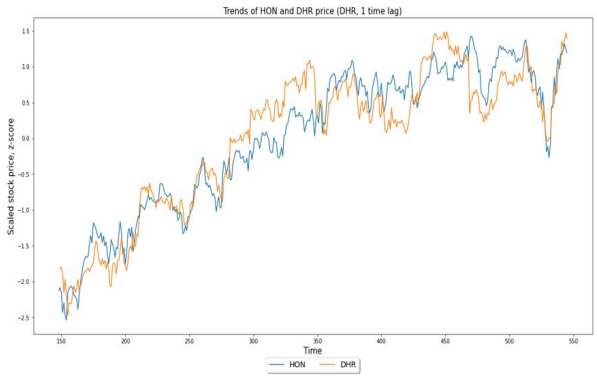
	Correlation matrix			HSIC matrix		
number of	Test Acc	F1 score	MCC	Test Acc	F1 score	MCC
cluster3	**0.5825**	0.5663	**0.1886**	0.5533	0.4871	0.1175
cluster4	0.5765	0.5463	0.1525	0.5644	0.4869	0.1178
cluster5	0.5623	0.5423	0.1454	0.5613	0.5047	0.1181
cluster8	0.5686	**0.5797**	0.1299	0.5613	0.4961	0.1119

Table 2.6: Comparison in 2 affinity matrices. 3 number of clusters outperform other # of cluster.

	Correlation			HSIC		
Top s neighors	Acc	F1	Mcc	Acc	F1	Mcc
10	0.5669	0.5065	0.1170	0.5548	0.5124	0.1024
20	0.5643	0.5161	0.1273	0.5577	0.5254	0.1244
30	0.5612	0.5205	0.1264	0.5579	0.5123	0.1232
40	0.5462	0.4704	0.1253	0.5767	0.5917	0.1219

Table 2.7: Comparison in 2 affinity matrices. Selecting top 40 neighbors outperform with respect to Acc and F1 score.

We also compare the performance with selections s neighbors. Table 2.7 show the performance with 2 types affinity matrices. Selecting top 40 neighbors outperform other number of neighbors with respect to Acc and F1 score. From table 2.7 and 2.6, we need to investigate causality between 2 stocks. We adopt Granger causality method. We check causality for 87



A stock	B stock	A stock industry	B stock industry
D	PPL	Utilities	Utilities
D	AEP	Utilities	Utilities
SRE	PPL	Utilities	Utilities
XOM	SLB	Energy	Energy
BSAC	BCH	Financials	Financials
UL	UN	Consumer	Consumer
PTR	SNP	Energy	Energy
PPL	D	Utilities	Utilities
SO	D	Utilities	Utilities
AEP	D	Utilities	Utilities
DHR	HON	Healthcare	Industrials
PCG	D	Utilities	Utilities
UN	UL	Consumer	Consumer
DUK	D	Utilities	Utilities

Figure 2.17: (Left) price trends of Danaher and Honeywell company, (Right) Stock lists of bi-directional causalities

stocks and there are 11 bi-directional causalities. Although most of the stocks are same industry, DHR (Danaher), HON (Honeywell) are not same industry. However, the stock trends are highly correlated (see fig. 2.17). But we can find clues in the financial report for 2 companies, 10-K in 2015. DHR and HON are similar market caps (44th and 48th, respectively). Danaher produces retail, commercial, petroleum, environmental monitoring and leak detection system. In case of HON produce Refinery materials (equipment) and consulting services to efficiently produce petroleum. we can see Honeywell is producing parts that require Danaher.

2.5.2 Market simulation

To further evaluate the performance of our proposed method for extreme market prediction, we simulate a prediction-based trading to test whether the predictions made by the methods can make profit. We test our performance real stock trading in a virtual market simulator and follow

Dyckman [46]'s strategy, which copies the behavior that a virtual market participants use our model to get earns in a simple way. If the model predicts that the price of the stock will increase the next step, the virtual trader will take a long position, vice versa. All strategic returns in this section are calculated as transaction costs and slippage, and in the real world, we can focus on the predictive power of the model itself. Thus, We set transaction costs (taxes, commissions) as \$0.00311 per a stock and slippage as 0.15% for each transaction. We follow rules of U.S. Securities and Exchange Commission (SEC) and Goldman Sachs [47, 48]. Cumulative rate of return is the aggregate amount that the investment has gained or lost over time, independent of the period of time involved. Accuracy of models can only measure the ability of the classification-based prediction, which correspond to ranges of future return, while what actually matters in market practice is the profitability, which is correlated to the amount of rise or fall.For example, profit made by two correctly predicted samples maybe absorbed by loss caused by one incorrectly predicted sample, if the actual amount of the rise or fall in the future of the incorrectly predicted sample is sufficiently large. cumulative returns can be formulated as follow:

$$\frac{\sum_{n=0}^{n}\left(\Pi_{t=0}^{t}\frac{Rtn_{t+1,n}}{Rtn_{t,n}}-1\right)}{total number of stock}, Rtn_t = \frac{\text{Price}_{t,n}}{\text{Price }_{t-1,n}} \tag{II.6}$$

When our proposed method well predict, Returns can be calculated as positive ratio, vice versa. After that, we product the return ratios step by step and record those. We simulate virtual investment during 3 months (Oct.02.2015 $\sim$ Jan.01.2016).

To compare against conventional momentum strategies in the finance market, we also adopt the following benchmarks:

- Long or short only strategies means that output of prediction is all positive (1, up) or negative (-1, down).
- Randomly chosen ground truth is randomly buy and sell strategy
- Moving Average Convergence Divergence (MACD) indicator is a trading indicator used in technical analysis of stock prices, created by Gerald Appel [49]. It is designed to reveal changes in the strength, direction, momentum, and duration of a trend in a stock's price.

In Fig. 2.18, our proposed method outperform both previous *state-the-of-art* method and technical strategy and baseline, Adv-ALSTM. Also, cumulative returns for random, all long and short strategies is less than 1.00%. In addition, we can see from 2.18 that all prediction-based simulations are significantly more profitable than the randomly buy and sell strategy We can see that referring to the cumulative rate of return chart for performance comparison to show

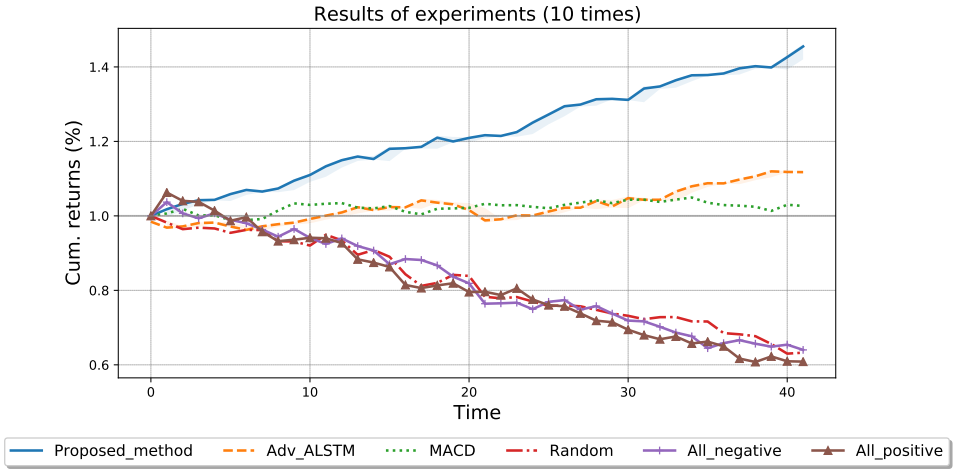


Figure 2.18: Cumulative Returns

the effect of predicted performance on 1.6 % improvement. However, Adv-LSTM and MACD strategies yield 1.1% and 1.04, respectively. Thus enhancing prediction performance as 1.6% is meaningful. It implies that prediction models involved can capture suitable trading points to make profits. Among these prediction models,all simulations based on predictions from machine learning and deep learning models result in better returns than others. If we have 10 million dollar, out excess return is approximately $600,000. Knowing the real rate of return of investments is also important. We can conclude from these results that models using deep learning methodologies have better capabilities of capturing profitable and stable signals than traditional methods.

CHAPTER III

Conclusion

We show that hierarchical clustering is meaningful for enhancing stock prediction performance by selecting a topological space for clusters of stocks traded on a stock market. It also describes the research of affective factors, by defining a specific group of stocks. The space and the hierarchical structure regarded with it, is achieved by using information on historical stock price only. This result means that historical stock prices have valuable and detectable economic information. Also, our proposed method outperforms Adv-ALSTM. Besides, the performance by considering both qualitative and quantitative features in financial reports is better than that of only considering the qualitative or quantitative features. In the future, we would like to investigate the effect of other industry detail features of global factors and industry instead of using historical dataset and local feature. Finally, in the perspective of the predicting direction of stock price, it will be worth to adopt the representative feature to make a classifier in the future.

CHAPTER IV

Acknowledgement

I would like to express my gratitude to my supervisor Kwang In Kim for the useful comments, remarks and engagement through the learning process of this master thesis. Furthermore, I am gratefully indebted to him for his very valuable comments on this thesis. I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. I will be grateful forever for your love.

Appendix

Ticker	Name	Industries
AAPL	Apple Inc.	Information Technology
ABBV	AbbVie Inc.	Healthcare
AEP	American Electric Power	Utilities
AMGN	Amgen Inc.	Healthcare
AMZN	Amazon.com Inc.	Consumer goods
BA	Boeing Company	Industrials
BAC	Bank of America Corp	Financials
C	Citigroup Inc.	Financials
CAT	Caterpillar Inc.	Industrials
CELG	Celgene Corp.	Healthcare
CHTR	Charter Communications	Consumer goods
CMCSA	Comcast Corp.	Consumer goods
CSCO	Cisco Systems	Information Technology
CVX	Chevron Corp.	Energy
D	Dominion Energy	Utilities
DHR	Danaher Corp.	Healthcare
DIS	The Walt Disney Company	Consumer goods
DUK	Duke Energy	Utilities
EXC	Exelon Corp.	Utilities
FB	Facebook, Inc.	Information Technology
GD	General Dynamics	Industrials
GE	General Electric	Industrials

GOOG	Alphabet Inc Class C	Information Technology
HD	Home Depot	Consumer goods
HON	Honeywell Int'l Inc.	Industrials
INTC	Intel Corp.	Information Technology
JNJ	Johnson & Johnson	Healthcare
JPM	JPMorgan Chase & Co.	Financials
KO	Coca-Cola Company (The)	Consumer goods
LMT	Lockheed Martin Corp.	Industrials
MA	Mastercard Inc.	Information Technology
MCD	McDonald's Corp.	Consumer goods
MDT	Medtronic plc	Healthcare
MMM	3M Company	Industrials
MO	Altria Group Inc	Consumer goods
MRK	Merck & Co.	Healthcare
MSFT	Microsoft Corp.	Information Technology
NEE	NextEra Energy	Utilities
ORCL	Oracle Corp.	Information Technology
PCG	PG&E Corp.	Utilities
PEP	PepsiCo Inc.	Consumer goods
PFE	Pfizer Inc.	Healthcare
PG	Procter & Gamble	Consumer goods
PM	Philip Morris International	Consumer goods
PPL	PPL Corp.	Utilities
SLB	Schlumberger Ltd.	Energy
SO	Southern Co.	Utilities
SRE	Sempra Energy	Utilities
T	AT&T Inc.	Telecommunication Services
UNH	United Health Group Inc.	Healthcare
UPS	United Parcel Service	Industrials
UTX	United Technologies	Industrials
V	Visa Inc.	Information Technology
VZ	Verizon Communications	Telecommunication Services
WFC	Wells Fargo	Financials
WMT	Wal-Mart Stores	Consumer goods
XOM	Exxon Mobil Corp.	Energy
ABB	ABB Ltd	Industrials
AGFS	AgroFresh	Consumer goods
BABA	Alibaba	Consumer goods

BBL	BHP	Basic Materials
BCH	Banco de Chile	Financials
BHP	BHP Group	Basic Materials
BP	BP p.l.c.	Energy
BRK-A	Berkshire Hathaway	Financials
BSAC	Banco Santander-Chile	Financials
BUD	Anheuser-Busch	Consumer goods
CHL	China Mobile	Telecommunication Services
CODI	Compass Diversified	Industrials
HRG	Heritage NOLA Bancorp	Financials
HSBC	HSBC Holdings	Financials
IEP	Icahn Enterprise	Industrials
NGG	National Grid	Utilities
NVS	Novartis AG	Healthcare
PCLN	Booking Holding	Consumer goods
PICO	PICO Holdings	Utilities
PTR	PetroChina	Energy
RDS-B	Royal Dutch Shell	Energy
REX	REX American Resources	Energy
SNP	China Petroleum	Energy
SNY	Sanofi	Healthcare
SPLP	Steel Partners Holdings	Industrials
TM	Toyota	Consumer goods
TOT	TOTAL	Energy
TSM	Taiwan Semiconductor Manufacturing	Information Technology
UL	The Unilever Group (UK)	Consumer goods
UN	The Unilever Group (Netherlands)	Consumer goods

Table 4.1: Lists of companies. Ticker is a symbol that arrangements of characters representing particular securities listed on an exchange or otherwise traded publicly.

We adopt Industry Classification Benchmark (ICB) methods by Dow Jones Index (DJI). The ICB is classified into 10 industries and 18 super-sectors. Our dataset consist of 9 industries and 14 sectors. To simplify the problem, we utilize only 9 industries information.

References

[1] James W Hall, "Adaptive selection of us stocks with neural nets," *Trading on the edge: neural, genetic, and fuzzy systems for chaotic financial markets. New York: Wiley*, pp. 45–65, 1994. 1

[2] Tao Li, Qi Li, Shenghuo Zhu, and Mitsunori Ogihara, "A survey on wavelet applications in data mining," *ACM SIGKDD Explorations Newsletter*, vol. 4, no. 2, pp. 49–68, 2002. 1

[3] Burton G Malkiel and Eugene F Fama, "Efficient capital markets: A review of theory and empirical work," *The journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970. 1

[4] Mark Coleman, "Cointegration-based tests of daily foreign exchange market efficiency," *Economics Letters*, vol. 32, no. 1, pp. 53–59, 1990. 1

[5] Craig S Hakkio and Mark Rush, "Market efficiency and cointegration: an application to the sterling and deutschemark exchange markets," *Journal of international money and finance*, vol. 8, no. 1, pp. 75–88, 1989. 1

[6] John P Lajaunie, Bruce L McManis, and Atsuyuki Naka, "Further evidence on foreign exchange market efficiency: An application of cointegration tests," *Financial Review*, vol. 31, no. 3, pp. 553–564, 1996. 1

[7] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung, *Time series analysis: forecasting and control*, John Wiley & Sons, 2015. 2

[8] Lazar Dorina and Ureche Simina, "Testing efficiency of the stock market in emerging economies," *The Journal of the Faculty of Economics-Economic Science Series*, vol. 2, pp. 827–831, 2007. 2

[9] Adebiyi A Ariyo, Adewumi O Adewumi, and Charles K Ayo, "Stock price prediction using the arima model," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. IEEE, 2014, pp. 106–112. 2

[10] Ping-Feng Pai and Chih-Sheng Lin, "A hybrid arima and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005. 2

[11] Alejandro Justiniano and Giorgio E Primiceri, "The time-varying volatility of macroeconomic fluctuations," *American Economic Review*, vol. 98, no. 3, pp. 604–41, 2008. 2

[12] Sang Bin Lee and Ki Yool Ohk, "Stock index futures listing and structural change in time-varying volatility," *Journal of Futures Markets*, vol. 12, no. 5, pp. 493–509, 1992. 2

[13] George S Atsalakis and Kimon P Valavanis, "Surveying stock market forecasting techniques–part ii: Soft computing methods," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5932–5941, 2009. 2

[14] Md Rafiul Hassan, Baikunth Nath, and Michael Kirley, "A fusion model of hmm, ann and ga for stock market forecasting," *Expert systems with Applications*, vol. 33, no. 1, pp. 171–180, 2007. 2

[15] Ritika Singh and Shashi Srivastava, "Stock prediction using deep learning," *Multimedia Tools and Applications*, vol. 76, no. 18, pp. 18569–18584, 2017. 2

[16] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins, "Learning to forget: Continual prediction with lstm," 1999. 2

[17] Kai Chen, Yi Zhou, and Fangyan Dai, "A lstm-based method for stock returns prediction: A case study of china stock market," in *2015 IEEE international conference on big data (big data)*. IEEE, 2015, pp. 2823–2824. 2, 3, 22

[18] Thomas Fischer and Christopher Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018. 2

[19] Wei Bao, Jun Yue, and Yulei Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PloS one*, vol. 12, no. 7, 2017. 2

[20] M Hiransha, E Ab Gopalakrishnan, Vijay Krishna Menon, and KP Soman, "Nse stock market prediction using deep-learning models," *Procedia computer science*, vol. 132, pp. 1351–1362, 2018. 2

[21] Ming-Chi Lee, "Using support vector machine with a hybrid feature selection method to the stock trend prediction," *Expert Systems with Applications*, vol. 36, no. 8, pp. 10896–10904, 2009. 2

[22] Li Zhang, Fulin Wang, Bing Xu, Wenyu Chi, Qiongya Wang, and Ting Sun, "Prediction of stock prices based on lm-bp neural network and the estimation of overfitting point by rdci," *Neural Computing and Applications*, vol. 30, no. 5, pp. 1425–1444, 2018. 2

[23] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015. 2

[24] Christian Häger and Henry D Pfister, "Nonlinear interference mitigation via deep neural networks," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*. IEEE, 2018, pp. 1–3. 2

[25] Mohammad Mekayel Anik, Mohammad Shamsul Arefin, and M Ali Akber Dewan, "An intelligent technique for stock market prediction," in *Proceedings of International Joint Conference on Computational Intelligence*. Springer, 2020, pp. 721–733. 3

[26] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis, "Using deep learning for price prediction by exploiting stationary limit order book features," *Applied Soft Computing*, p. 106401, 2020. 3

[27] Robert D Edwards, WHC Bassetti, and John Magee, *Technical analysis of stock trends*, CRC press, 2007. 3

[28] Robert P Schumaker and Hsinchun Chen, "A quantitative stock prediction system based on financial news," *Information Processing & Management*, vol. 45, no. 5, pp. 571–583, 2009. 3

[29] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan, "Deep learning for event-driven stock prediction," in *Twenty-fourth international joint conference on artificial intelligence*, 2015. 3

[30] Robert P Schumaker and Hsinchun Chen, "Textual analysis of stock market prediction using breaking financial news: The azfin text system," *ACM Transactions on Information Systems (TOIS)*, vol. 27, no. 2, pp. 1–19, 2009. 3

[31] Richard A Johnson, Irwin Miller, and John E Freund, *Probability and statistics for engineers*, vol. 2000, Pearson Education London, 2000. 8

[32] Michael W McCracken, "Asymptotics for out of sample tests of granger causality," *Journal of econometrics*, vol. 140, no. 2, pp. 719–752, 2007. 9

[33] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf, "Measuring statistical dependence with hilbert-schmidt norms," in *International conference on algorithmic learning theory*. Springer, 2005, pp. 63–77. 11, 12

[34] Alfréd Rényi, "On measures of dependence," *Acta Mathematica Academiae Scientiarum Hungarica*, vol. 10, no. 3-4, pp. 441–451, 1959. 12

[35] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua, "Enhancing stock movement prediction with adversarial training," *arXiv preprint arXiv:1810.09936*, 2018. 13, 20, 21, 22

[36] Chris Cheadle, Yoon S Cho-Chung, Kevin G Becker, and Marquis P Vawter, "Application of z-score transformation to affymetrix data.," *Applied bioinformatics*, vol. 2, no. 4, pp. 209–217, 2003. 14

[37] Rosario Nunzio Mantegna, "Degree of correlation inside a financial market," in *AIP Conference Proceedings*. American Institute of Physics, 1997, vol. 411, pp. 197–202. 15

[38] Jenna Birch, Athanasios A Pantelous, and Kimmo Soramäki, "Analysis of correlation based networks representing dax 30 stock price returns," *Computational Economics*, vol. 47, no. 4, pp. 501–525, 2016. 15

[39] Joshua Abor, "Industry classification and the capital structure of ghanaian smes," *Studies in Economics and Finance*, 2007. 16

[40] Yumo Xu and Shay B Cohen, "Stock movement prediction from tweets and historical prices," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1970–1979. 20

[41] David MQ Nelson, Adriano CM Pereira, and Renato A de Oliveira, "Stock market's price movement prediction with lstm neural networks," in *2017 International joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1419–1426. 20

[42] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017. 20, 22

[43] Cyril Goutte and Eric Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *European Conference on Information Retrieval*. Springer, 2005, pp. 345–359. 21

[44] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari, "Optimal classifier for imbalanced data using matthews correlation coefficient metric," *PloS one*, vol. 12, no. 6, 2017. 21

[45] Michele Tumminello, Fabrizio Lillo, and Rosario N Mantegna, "Correlation, hierarchies, and networks in financial markets," *Journal of economic behavior & organization*, vol. 75, no. 1, pp. 40–58, 2010. 22

[46] Thomas Dyckman, Donna Philbrick, and Jens Stephan, "A comparison of event study methodologies using daily stock returns: A simulation approach," *Journal of Accounting Research*, pp. 1–30, 1984. 24

[47] Jack D Schwager, *Market wizards, updated: Interviews with top traders*, John Wiley & Sons, 2012. 24

[48] Jack D Schwager, *The new market wizards: Conversations with America's top traders*, vol. 95, John Wiley & Sons, 2012. 24

[49] Jamil Baz, Nicolas Granger, Campbell R Harvey, Nicolas Le Roux, and Sandy Rattray, "Dissecting investment strategies in the cross section and time series," *Available at SSRN 2695101*, 2015. 24